FYP – Rough Plan   
  
For your "Wardrobe Buddy" app, here's a step-by-step guide to implementing it, including the choice of languages, frameworks, and tasks to approach your final year project:

**1. Choose the Technology Stack**

* **Mobile App Development**:
  + **Frontend**: Use **Flutter** (Dart) or **React Native** (JavaScript/TypeScript) for a cross-platform mobile application. These frameworks will allow you to create both iOS and Android apps with a single codebase.
  + **Backend**: Use **Node.js** (JavaScript) with **Express** for server-side logic and handling user login, data storage, etc. Alternatively, you can use **Django** (Python) or **Flask** if you're comfortable with Python.
  + **Database**: Use **Firebase** or **MongoDB** for storing user data, images, and outfit combinations. Firebase works well with mobile apps and includes built-in authentication and storage.
  + **Computer Vision**: Use **Python** with libraries like **OpenCV** and **TensorFlow** to handle image processing and background removal.

**2. Step-by-Step Development Plan**

**Phase 1: Set Up Project Environment**

* Install the necessary software and tools:
  + Flutter/React Native for mobile app development.
  + Node.js or Python for backend.
  + Firebase SDK for authentication and database management.
  + Python environment for Computer Vision (OpenCV, TensorFlow).
* Create a **version control repository** (GitHub/GitLab).

**Phase 2: Basic User Authentication**

* Implement a **user login and registration** system using Firebase Authentication or a custom backend (Node.js with MongoDB or Django with SQL).
* Store basic user data (name, email, saved outfits, etc.).

**Phase 3: Photo Upload & Background Removal**

* **Photo Capture**: Allow the user to take or upload photos of their clothes.
  + **Remove background** from each photo using Python (OpenCV). Implement an API using **Flask** or **FastAPI** to process images on the backend.
* **Split images** into 4 sections (headwear, body, legs, footwear).
* Save these images in Firebase Storage or the database.

**Phase 4: Display Clothes and Allow Toggling**

* Develop the UI for showing the user’s wardrobe, categorized into headwear, body, legs, and footwear.
* For shoes, enable the user to view **different angles** (inside, outside, top-down) by toggling between images.

**Phase 5: Computer Vision for Outfit Generation**

* Implement a **Color Recognition Algorithm** using TensorFlow or scikit-learn.
  + Train the model to suggest outfits based on color combinations.
* Implement the **rule-based system** for outfit generation:
  + If a T-shirt has a graphic, avoid matching it with graphic trousers.
  + Allow users to add layers like jumpers or outwear.
* Implement the "sandwich method" or other style rules in the future by expanding the algorithm.

**Phase 6: Outfit Display & Interaction**

* Display the generated outfits to the user.
  + Provide a **front and back view** for each clothing item.
* Allow the user to "favourite" an outfit, which saves it to their list.
* Implement an option to **regenerate outfits** with a rule to avoid showing the same combination twice.

**Phase 7: Favourites and Outfit Management**

* Implement a **Favourites screen** where users can view saved outfits.
* Add options to **delete** or **edit** saved outfits.

**Phase 8: Social Features & Blog Integration**

* Add links to popular fashion blogs (e.g., Hypebeast, Basement, Vogue).
* Implement a basic **social feature** where users can follow each other and view each other's saved outfits (use Firebase Firestore or MongoDB for storing relationships).
* Allow users to **share** outfits with friends within the app.

**3. Project Timeline**

|  |  |
| --- | --- |
| **Task** | **Time Estimate** |
| Set up environment and authentication | 1 week |
| Photo upload and background removal | 2 weeks |
| Categorize and store clothing items | 1 week |
| Implement outfit generation algorithm | 3 weeks |
| Display outfit and user interaction | 2 weeks |
| Favourites and outfit management | 1 week |
| Social and blog integration | 2 weeks |
| Testing and refinement | 2 weeks |

**4. Testing**

* **Unit Testing**: For each component (e.g., login, photo processing, outfit generation).
* **Integration Testing**: Test the full flow from login to outfit generation and favourites.
* **User Testing**: Get feedback from users to improve the UI and the accuracy of the outfit generation.

**5. Deployment**

* Deploy the mobile app to **Google Play Store** and **Apple App Store**.
* Host the backend on platforms like **Heroku**, **Google Cloud**, or **AWS**.
* Use Firebase Hosting for static content if needed.

**6. Optional Future Enhancements**

* Introduce **AR (Augmented Reality)** to allow users to try on outfits.
* Use **ML-based trend prediction** to suggest outfits based on current fashion trends.
* Implement a recommendation system to suggest new clothing items from external retailers.

By following this roadmap, you can break down the project into manageable parts and stay on track for your final year project!